

Figure 1.

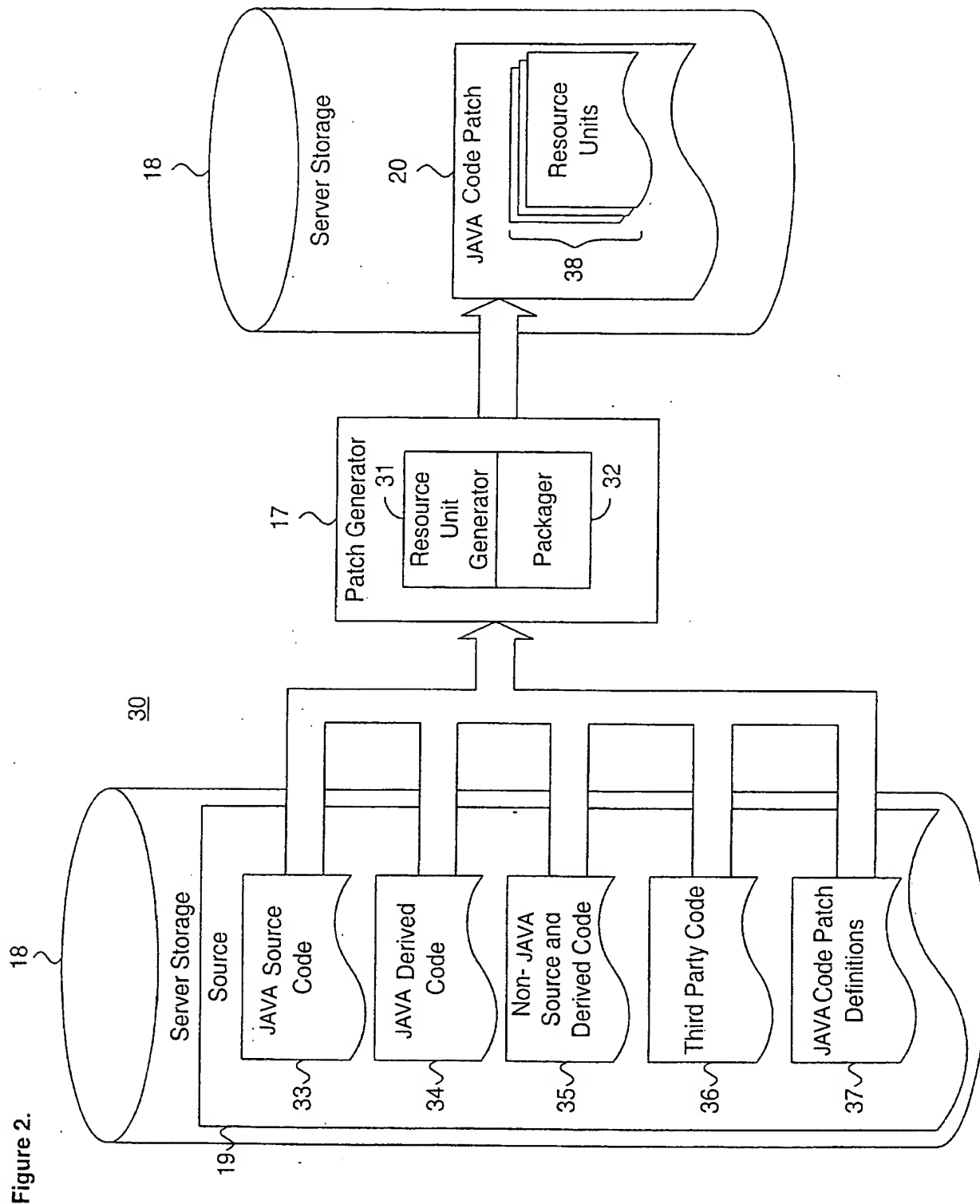


Title: SYSTEM AND METHOD FOR PROVIDING A JAVA CODE  
RELEASE STRUCTURE WITH GRANULAR CODE PATCHING

Inventor(s): Kenton E. Noble, et al.

Docket No. 50277-1955

Replacement Sheet 2 of 13



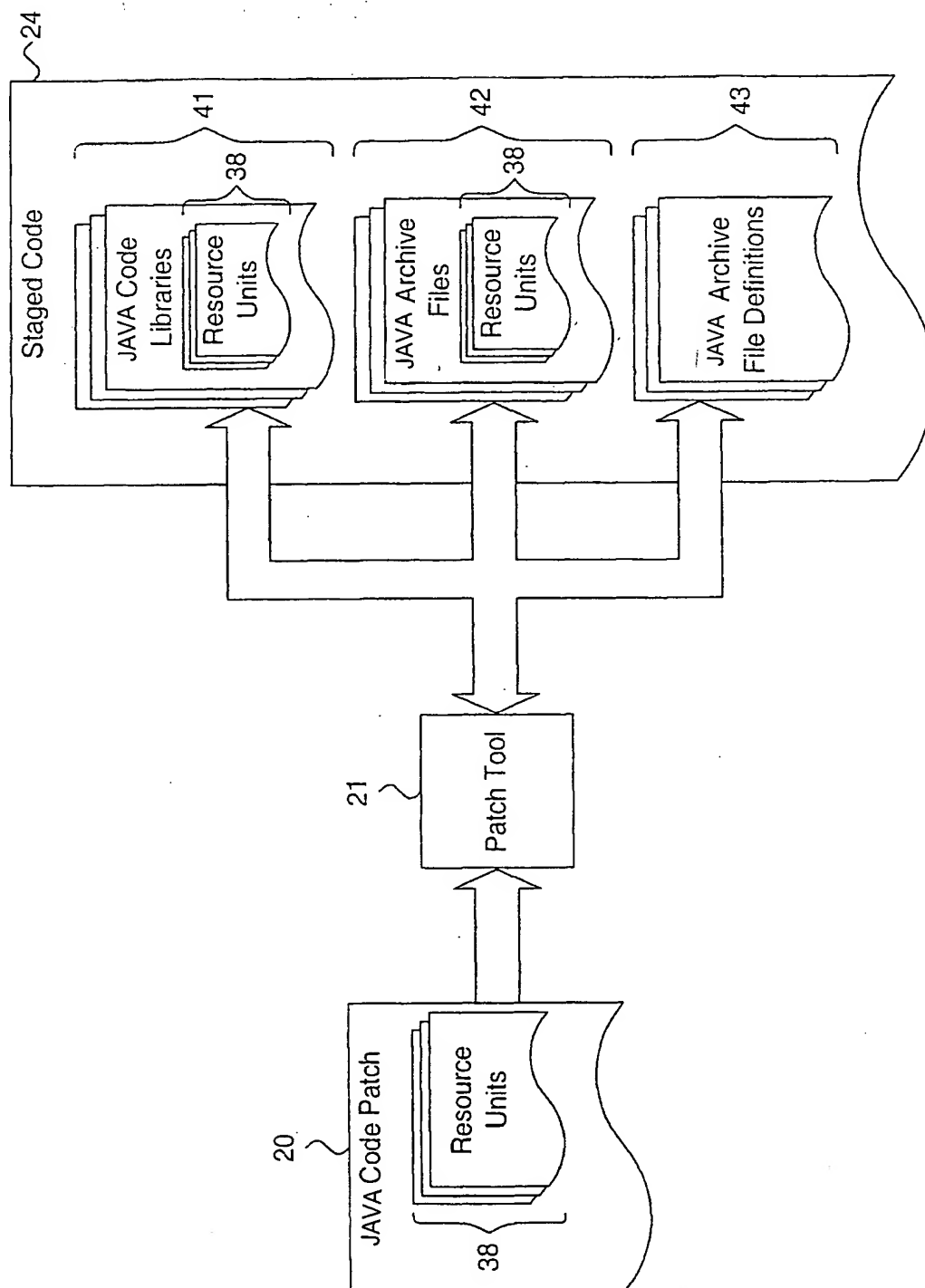
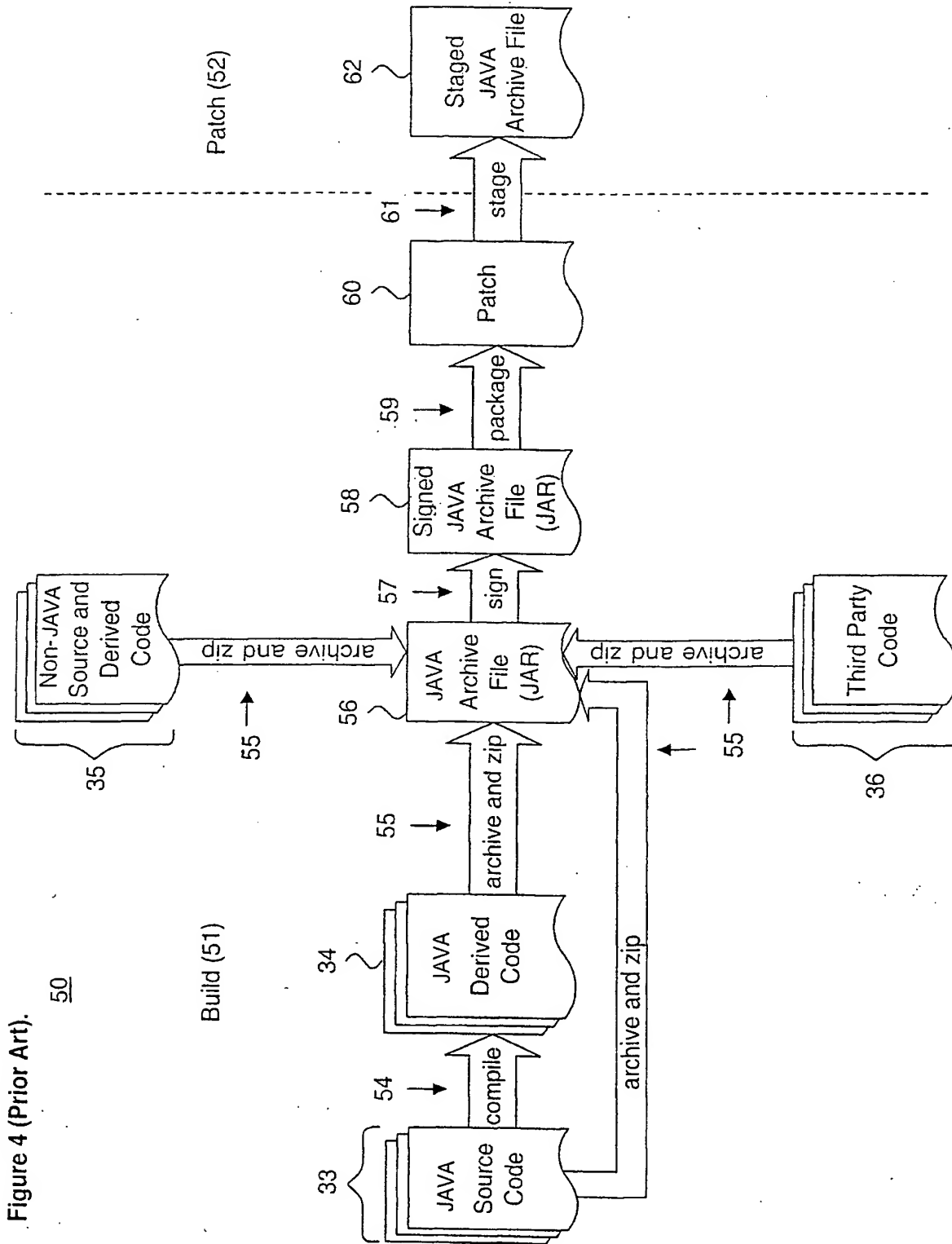


Figure 3.



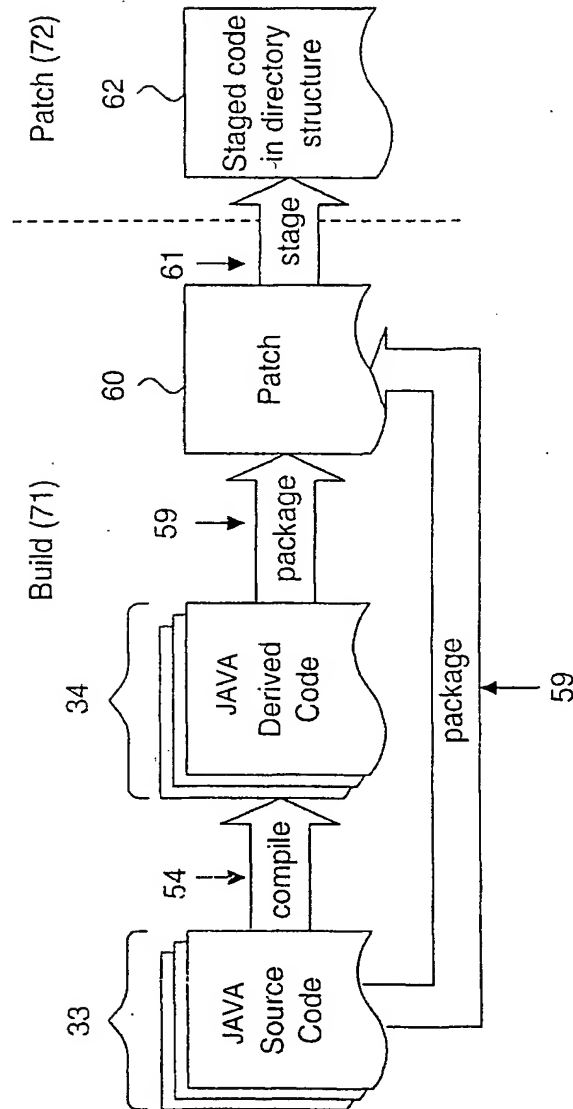
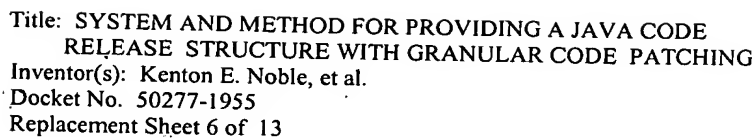


Figure 5 (Prior Art).



**Figure 6.**

The diagram illustrates a system architecture for generating Java code patches, divided into two main sections by a dashed line labeled "Patch (92)".

**Build (91) Section:**

- Path 99:** Starts with **JAVA Source Code** (33), which is processed through a **compile** step (54) to produce **JAVA Derived Code** (34). This code is then used to **generate** **Resource Units** (38).
- Path 100:** Starts with **Non-JAVA Source and Derived Code** (35), which is used to **generate** **Resource Units** (38).
- Integration:** The **JAVA Source Code** (33) and the **Resource Units** (38) from Path 100 are combined via a **generate** process to form a **Package** (20).

**Patch (92) Section:**

- The **Package** (20) is used to **compare** against existing **JAVA Code Libraries** (41).
- This comparison leads to a **compare and merge** process, resulting in a **JAVA Code Patch**.
- The patch is then used to **extract** information from the **JAVA Code Libraries** (41) to produce **JAVA Archive Files** (42).
- A separate path shows **Third Party Code** (36) being used to **generate** **Resource Units** (38), which are then integrated into the overall system.

**Legend:**

- 99: JAVA Source Code
- 100: Non-JAVA Source and Derived Code
- 90: Build (91)
- 92: Patch (92)

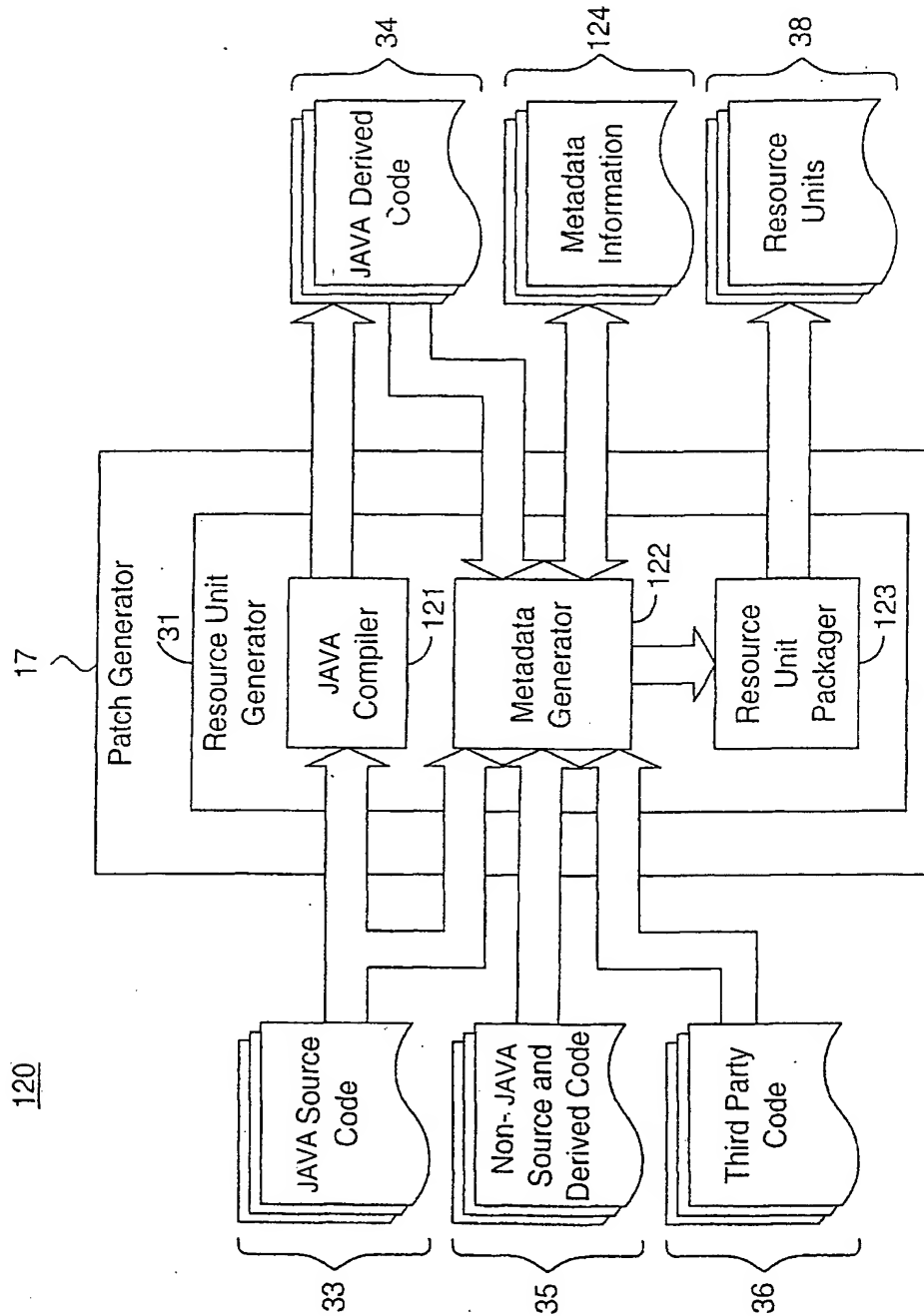


Figure 7.

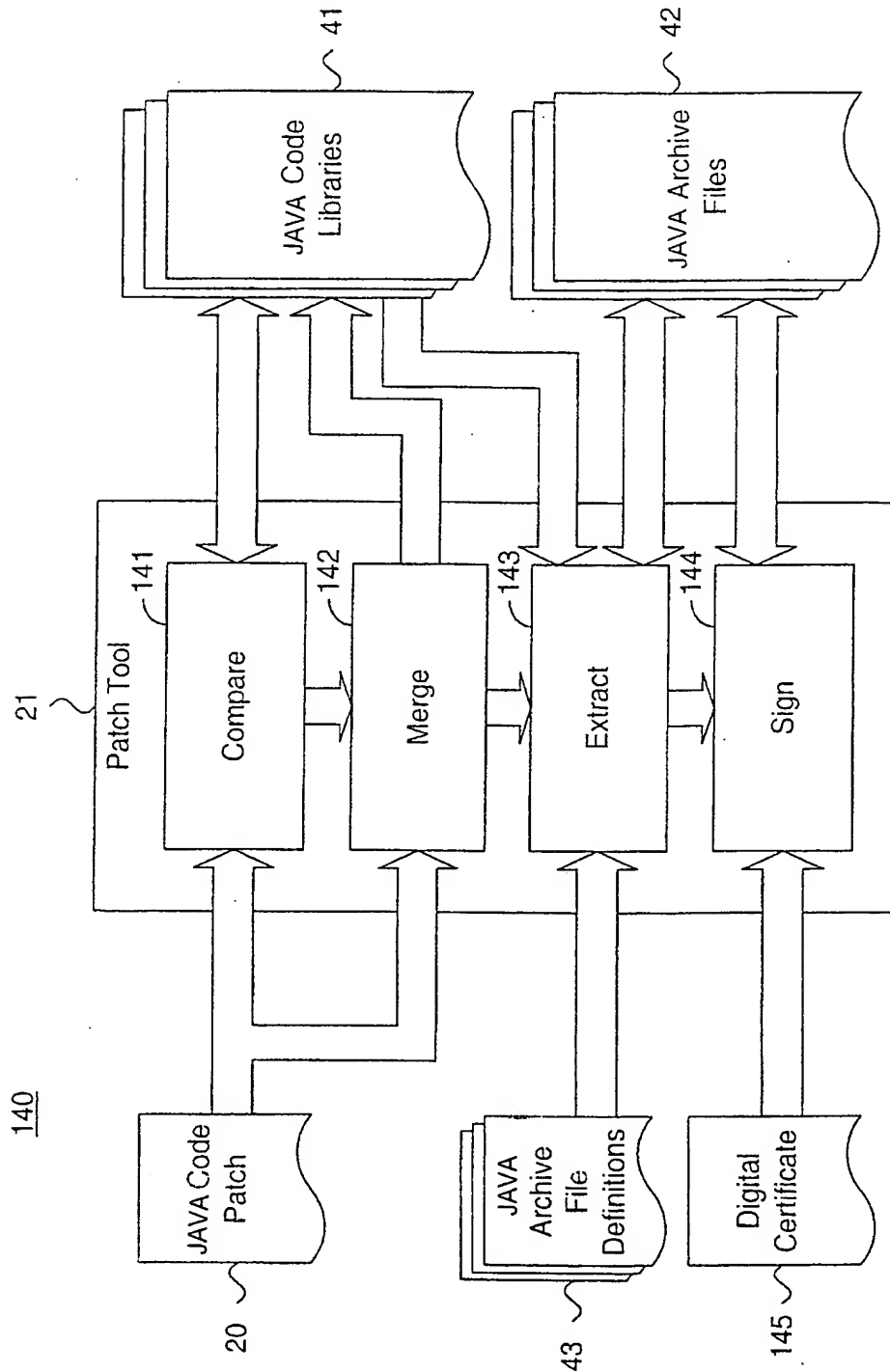


Figure 9.





Figure 10.

150

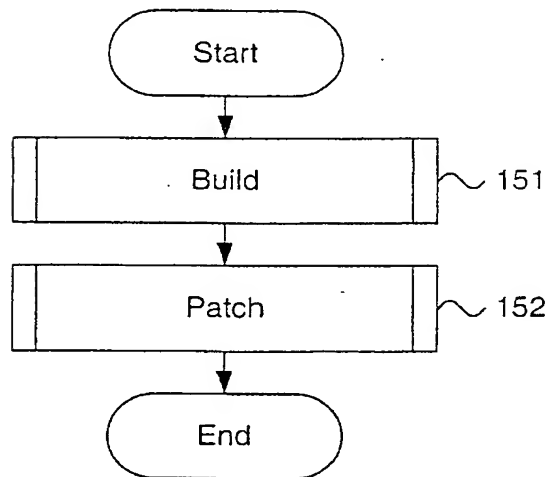


Figure 11.

160

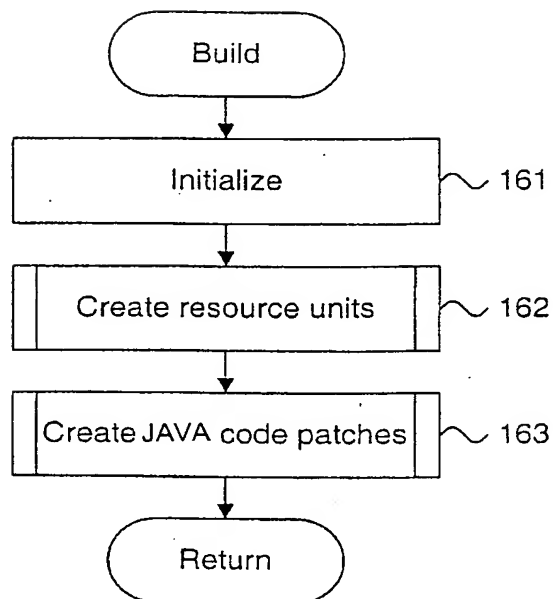




Figure 12.

170

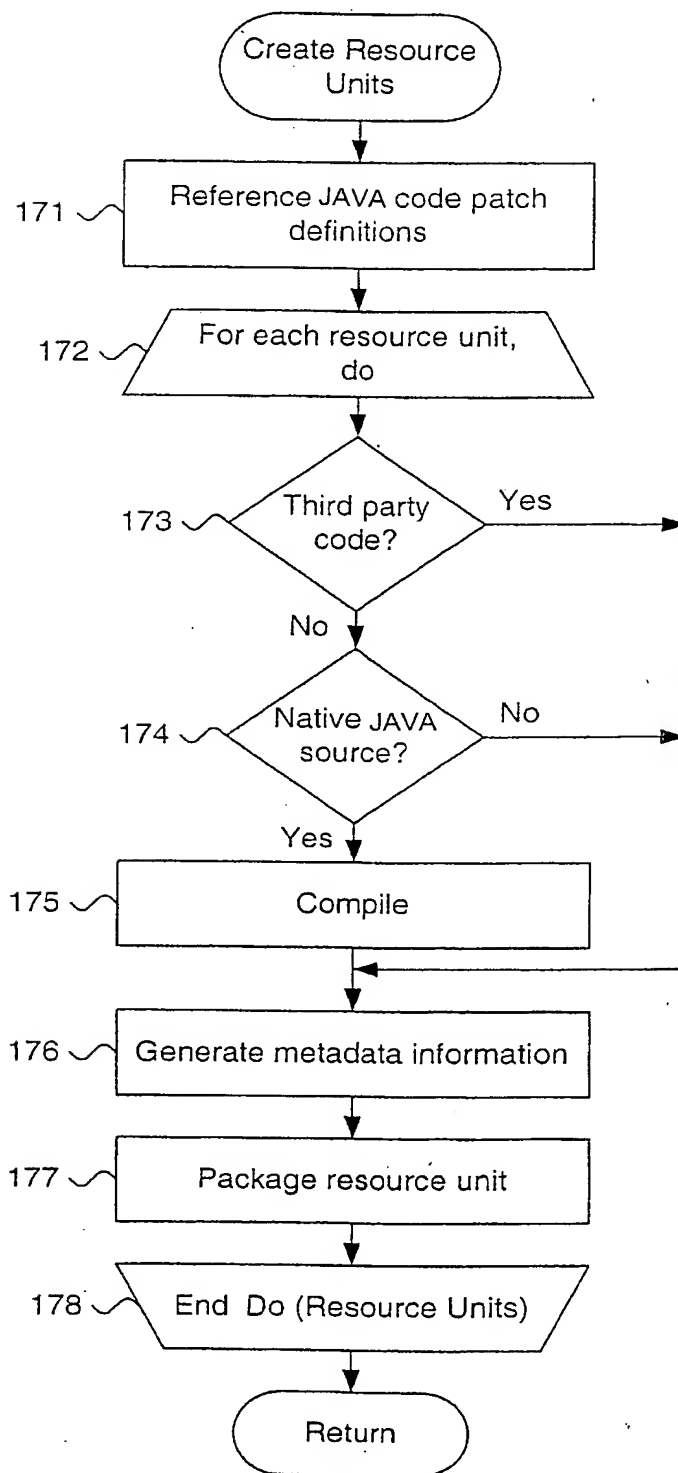




Figure 13.

180

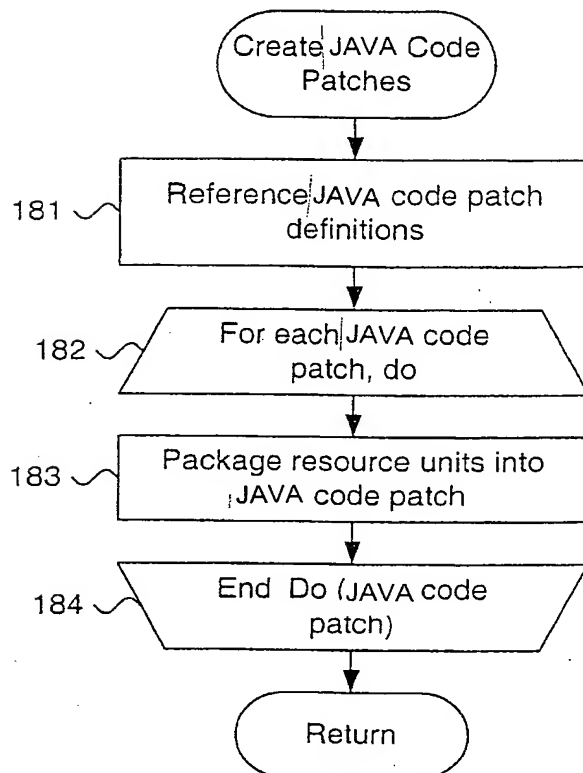


Figure 14.

190

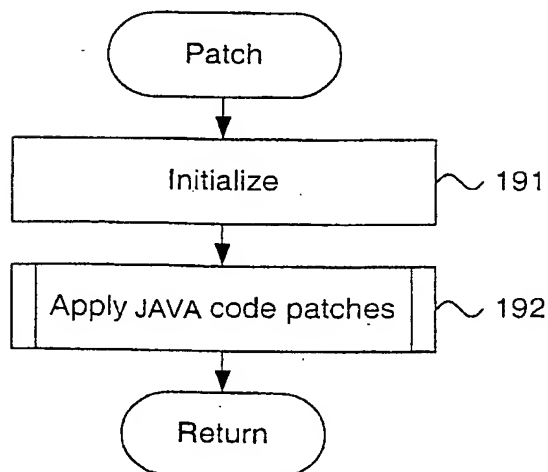




Figure 15.

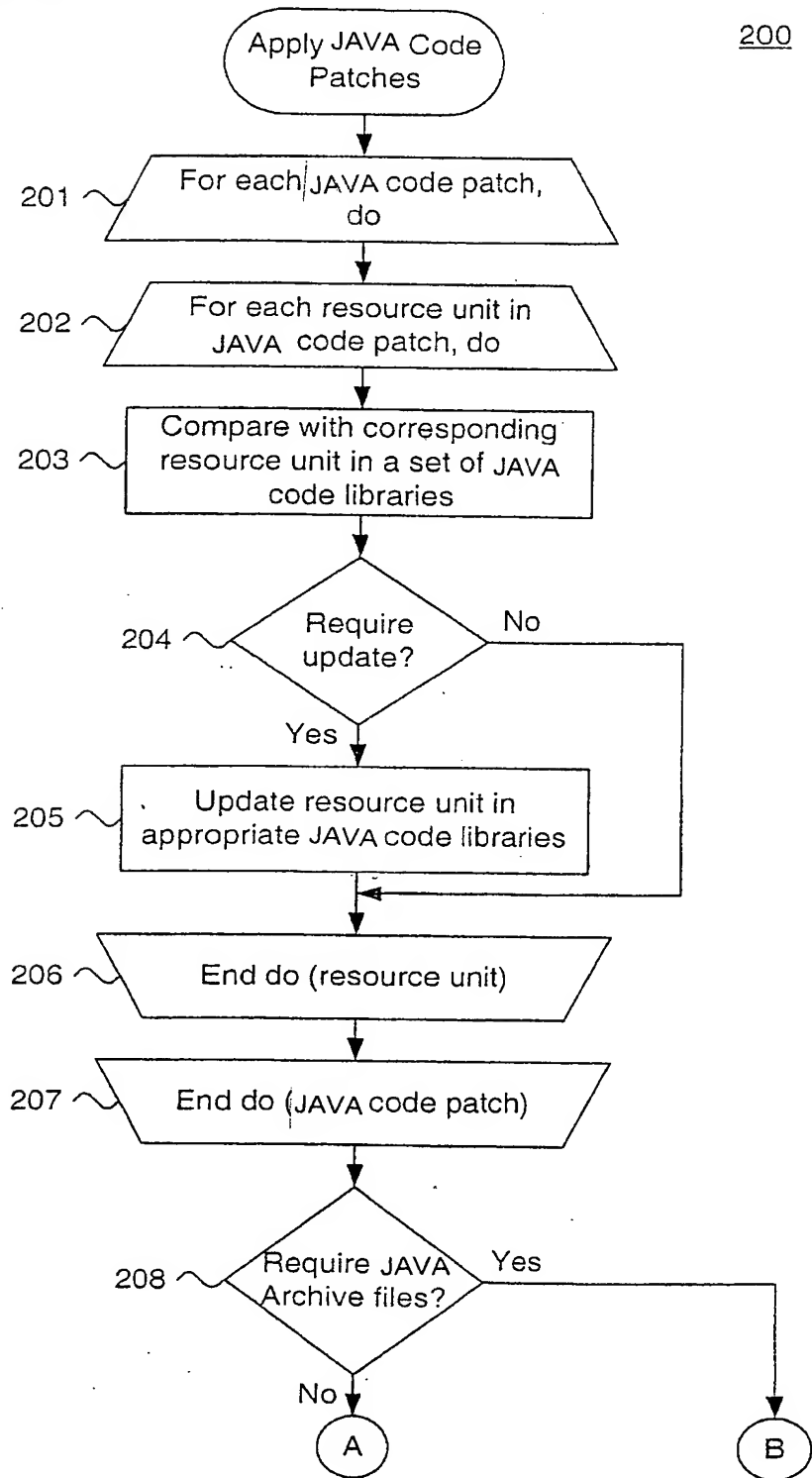




Figure 15 (con'd).

